Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/070501

A Recurrent Neural Network–based Forecasting System for Telecommunications Call Volume

Paris Mastorocostas[*], *Constantinos Hilas, Dimitris Varsamis and Stergiani Dova*

Department of Informatics and Communications, Technological Education Institute of Serres, 62124 Serres, Greece

Received: 27 Dec. 2012, Revised: 29 Apr. 2013, Accepted: 2 May. 2013
Published online: 1 Sep. 2013

Abstract: A recurrent neural network–based forecasting system for telecommunications call volume is proposed in this work. In particular, the forecaster is a Block–Diagonal Recurrent Neural Network with internal feedback. Model's performance is evaluated by use of real–world telecommunications data, where an extensive comparative analysis with a series of existing forecasters is conducted, including both traditional models as well as neural and fuzzy approaches.

Keywords: telecommunications data volume forecasting, neural modelling, block–diagonal recurrent neural network

1 Introduction

During the last twenty years the proliferation of mobile phones, along with the explosive growth in telecommunications services have turned this scientific field to an important and developing industry. Like any other business, carriers seek profit maximization and cost reduction, by managing network traffic effectively, by optimizing their infrastructure usage and by predicting future trends. In order to achieve their objectives, telecommunications managers and operation officers are in continuous need of accurate forecasting models of the call volume. In this perspective, this work aims at addressing the problem of call volume forecasting for the case of a large organization: a large Greek University is investigated, where new telephone numbers are added daily and a varying demand for outgoing trunks exists. The University's database stores the total number, as well as the number of the national, the international and the mobile calls per employee and per month. It has been noticed that the call classification into different categories reveals certain and different patterns between destinations.

Initially, traditional forecasting methods - mostly statistical - were applied to this particular problem. These methods can be divided into three major categories: The first method employed is the Naïve Forecast 1 [1], which uses the most recent observation as a forecast for the next time interval. A method that takes into account the seasonal factors is the Linear Extrapolation with Seasonal Adjustment (LESA, [2]): After seasonality has been removed from the original data, linear extrapolation is applied in order to forecast the future values of the series by employing the trend–cycle component. Finally, the projected trend–cycle component is adjusted, making use of the identified seasonal factors. LESA–M assumes multiplicative seasonality, while LESA–ADD is based on additive seasonality.

The second classic group of time–series analysis techniques includes the exponential smoothing methods, where a particular observation of the time series is expressed as a weighted sum of the previous observations. The weights for the previous data values constitute a geometric series and become smaller as the observations move further into the past. In processes without trend Simple Exponential Smoothing (SES) is applied, while linear trend and seasonal data are accommodated by Holt's [3] and Winters' [4] methods, respectively. Additionally, multiplicative seasonal models (Winters' MS) as well as additive seasonal models (Winters' AS) exist [5].

The last category concerns time series that exhibit damped trend, which refers to a regression component for the trend in the updating equation, expressed by means of a dampening factor. In these cases, some modifications of SES can be applied in order to deal with complex types of trend. An exponential smoothing model with damped trend and additive seasonality (DAMP AS) and its

* Corresponding author e-mail: mast@teiser.gr

multiplicative seasonality counterpart (DAMP MS) also exist, while a damped trend model on time series with no seasonality (DAMP NoS) can be fitted [5]. Finally, the Auto Regressive Integrated Moving Average method (ARIMA) and Seasonal ARIMA (SARIMA), which analyze stationary univariate time series data, have been very popular [6]. It presumes weak stationarity, equally spaced intervals or observations, and at least 30 to 50 observations. The well–established methods mentioned above have been studied in [2]. Linear models are also suggested by the ITU Recommendation E.507 for trend forecasting in telecommunications data [7]. It has been proved that Damped trend models and SARIMA are the most effective, when applied to the problem in hand, compared to the rest of the traditional methods.

Computational Intelligence methods were first applied in [8,9,10], where fuzzy and neurofuzzy systems were employed. In [8] a static fuzzy model is presented (OLS–FFM, Orthogonal Least Squares–based Fuzzy Forecasting Model), where an automated model–building process, based on the Orthogonal Least Squares algorithm, performs input selection, determines the number of fuzzy rules, which are of Takagi–Sugeno–Kang type [11], forms the consequent parts of the fuzzy rules and tunes the consequent parameters. Recurrent neurofuzzy forecasters are developed in [9] and [10]. In [9] the Locally Recurrent Neurofuzzy Forecasting System (LR–NFFS) is presented: the consequent part of each rule consist of a three–layer recurrent neural network, with internal feedback at the neurons of the hidden and output layers, and a single input common to the premise and consequent parts. In [10] the Recurrent Neurofuzzy Forecaster (ReNNFOR) is introduced, which is a respective recurrent system of reduced complexity, with unit feedback at the hidden layer's neurons. All three models exhibited significantly ameliorated prediction capabilities with respect to the traditional statistical methods, with the recurrent forecasters requiring only the value of the most recent observation, since they are capable of identifying the temporal relations via the internal feedback connection.

A sparse recurrent neural network, where the feedback connections are restricted to between pairs of state variables is suggested in [12], referred to as the Block–diagonal recurrent neural network (BDRNN). It is a two–layer network, a feedback layer comprising the block–diagonal links, and an output layer where the state variables are combined to formulate the network's output. The BDRNN architecture offers a significant feature: it constitutes a suitable choice for modelling nonlinear processes, where the overall dynamics can be decomposed into a set of lower–order dynamics. Under these conditions, the blocks of BDRNN can be used to model these sub–dynamics by adapting properly the respective block weights. In this perspective, a forecasting system, based on BDRNN, is proposed in this work. Its performance is compared with familiar forecasting approaches, like a series of seasonally adjusted linear extrapolation methods, Exponential Smoothing Methods, the SARIMA method, along with the aforementioned fuzzy and neural forecasters.

The rest of the paper is organized as follows: in Section 2, a brief presentation of the BDRNN is given and the modelling method is described. Section 3 hosts the experimental results and a comparative analysis, while the concluding remarks are given in Section 4.

2 The forecaster's structure and the modelling method

2.1 BDRNN

The block–diagonal recurrent neural network is a two–layer network. The hidden layer consists of pairs of neurons (blocks); there are feedback connections between the neurons of each pair, introducing dynamics to the network, while the output layer is static. Therefore, the BDRNN can be considered as a special form of locally–recurrent–globally–feedforward network [13,14]. The implementation of the proposed algorithm is straightforward, with no preconditions required. The configuration of BDRNN is presented in Fig. 1, where a single–input–single–output (SISO) BDRNN is shown, since a SISO network will be used in the sequel. For the sake of simplicity, a BDRNN with four blocks of neurons is shown.

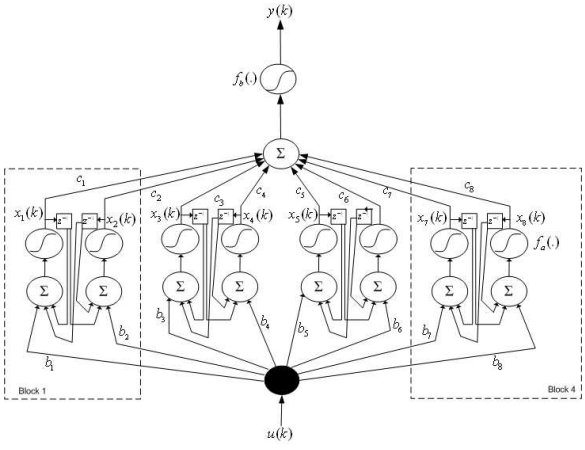


Fig. 1: Configuration of the Block–Diagonal Recurrent Neural Network.

The operation of a SISO BDRNN with N neurons at the hidden layer is described by the following set of state equations:

$$x(k) = \underline{f_a}\left(W \cdot x(k-1) + B \cdot u(k)\right) \tag{1a}$$

$$y(k) = f_b\left(C \cdot x(k)\right) \tag{1b}$$

where

– $\underline{f_a}$ is a N-element vector including the neuron activation functions of the hidden layer, and f_b is the activation function of the output layer. The activation functions are both chosen to be the sigmoid function $f(z) = \frac{1-e^{-2\cdot z}}{1+e^{-2\cdot z}}$.
– $u(k)$ is the input of the network at time k.
– $x(k) = [x_i(k)]$ is a N-element vector, comprising the outputs of the hidden layer. In particular, $x_i(k)$ is the output of the i-th hidden neuron at time k.
– $y(k)$ is the output of the network at time k.
– $B = [b_i]$ and $C = [c_i]$ are N-element input and output weight vectors, respectively.
– $W = [w_{i,j}]$ is the $N \times N$ block diagonal feedback matrix. In particular,

$$w_{i,j} = \begin{cases} \neq 0 \text{ if } i = j \\ \neq 0 \text{ if } i \neq j \text{ and } i = j-1 \text{ and } i \text{ is odd} \\ \neq 0 \text{ if } i \neq j \text{ and } i = j+1 \text{ and } i \text{ is even} \\ 0 \quad \text{otherwise} \end{cases}$$

The feedback matrix, W, is block diagonal: $W = diag\left[W^{(1)},...,W^{(N/2)}\right]$; each diagonal element, corresponding to a block of recurrent neurons, has a block submatrix in the form

$$W^{(i)} = \begin{bmatrix} w_{2i,2i} & w_{2i,2i+1} \\ w_{2i+1,2i} & w_{2i+1,2i+1} \end{bmatrix}, i = 1,2,...,\frac{N}{2} \quad (2)$$

The above equation describes the general case of BDRNN, called the BDRNN with free–form submatrices. A special case of BDRNN consists of scaled orthogonal submatrices taking the following form

$$W^{(i)} = \begin{bmatrix} w_{2i,2i} & w_{2i,2i+1} \\ -w_{2i,2i+1} & w_{2i,2i} \end{bmatrix} = \begin{bmatrix} w_i^{(1)} & w_i^{(2)} \\ -w_i^{(2)} & w_i^{(1)} \end{bmatrix}, i = 1,2,...,\frac{N}{2} \quad (3)$$

From (2) and (3) becomes evident that the Free–Form BDRNN consists of feedback submatrices with four distinct elements, thus providing a greater degree of freedom compared to the Scaled Orthogonal BDRNN, which has two weights at each feedback submatrix. Nevertheless, as discussed in [12], the latter network exhibits superior modelling capabilities than the Free–Form BDRNN.

In view of the above, the state equations (1) for the case of the Scaled Orthogonal BDRNN are written in the following form:

$$x_{2i-1}(k) = f_a\left(b_{2i-1}\cdot u(k) + w_i^{(1)}\cdot x_{2i-1}(k-1) + w_i^{(2)}\cdot x_{2i}(k-1)\right) \; i = 1,2,...,\frac{N}{2} \quad (4a)$$

$$x_{2i}(k) = f_a\left(b_{2i}\cdot u(k) - w_i^{(2)}\cdot x_{2i-1}(k-1) + w_i^{(1)}\cdot x_{2i}(k-1)\right) \; i = 1,2,...,\frac{N}{2} \quad (4b)$$

$$y(k) = f_b\left(\sum_{j=1}^{N} c_j\cdot x_j(k)\right) \quad (4c)$$

2.2 The modelling method

In literature there exist two learning schemes especially designed for this kind of neural network [12,15]. However, they are both designed for ensuring network stability: in [12] a simple gradient descent algorithm is used, with the addition of a feedfoward neural network which aims at keeps stable learning, operating in parallel. A similar task is performed more efficiently in [15] without the need for an additional network, by transforming the whole learning process to a constrained optimization problem. In this work the stability issue is not investigated, since the learning process has an adaptation scheme for the magnitude of the weight changes, thus confining the weight space.

The learning algorithm is based on **R**esilient Back**pro**pagation (RPROP, [16]). RPROP algorithm was originally developed for static neural networks and constitutes one of the best performing first order learning methods for neural networks, since it exhibits improved performance characteristics by remedying the drawbacks inherent to gradient descent learning [17]. A variation of RPROP for recurrent models is given in [18], where a similar behavior has been reported. In the case of BDRNN, the algorithm is modified in order to take into consideration the special features of the BDRNN, requiring calculation of the error gradients for the feedback weights.

Let us consider a training data set of k_f input–output pairs. The Mean Squared Error is selected as the error measure, where $y_d(k)$ is the actual observation:

$$E = \frac{1}{k_f}\cdot\sum_{k=1}^{k_f}(y(k)-y_d(k))^2 \quad (5)$$

$\frac{\partial^+ E(t)}{\partial\theta_l}$ and $\frac{\partial^+ E(t-1)}{\partial\theta_l}$ are the ordered derivatives [19] of E with respect to a network's weight θ_l ($\theta_l \in \left\{b_i, c_i, w_i^{(1)}, w_i^{(2)}\right\}$) at the present, t, and the preceding, $t-1$, epochs, respectively. The learning method is described as follows:

(a) For all weights θ_l, initialize the step sizes $\Delta_l^{(1)} = \Delta_0$
Repeat
(b) For all weights θ_l, compute the error gradient: $\frac{\partial^+ E(t)}{\partial\theta_l}$
(c) For all weights θ_l, update step sizes:
(c.1) If $\frac{\partial^+ E(t)}{\partial\theta_l}\times\frac{\partial^+ E(t-1)}{\partial\theta_l} > 0$ then

$$\Delta_l^{(t)} = \min\left\{\eta^+\cdot\Delta_l^{(t-1)}, \Delta_{\max}\right\} \quad (6a)$$

(c.2) Else if $\frac{\partial^+ E(t)}{\partial \theta_l} \times \frac{\partial^+ E(t-1)}{\partial \theta_l} < 0$ then

$$\Delta_l^{(t)} = \max\left\{\eta^- \cdot \Delta_l^{(t-1)}, \Delta_{\min}\right\} \quad (6b)$$

(c.3) Else

$$\Delta_l^{(t)} = \Delta_l^{(t-1)} \quad (6c)$$

(d) Update weights θ_l:

$$\theta_l(t+1) = \theta_l(t) + \Delta\theta_l(t) = \theta_l(t) - sign\left(\frac{\partial^+ E(t)}{\partial \theta_l}\right) \cdot \Delta_l^{(t)} \quad (7)$$

Until convergence

where the step sizes are bounded by $\Delta_{\min}$, $\Delta_{\max}$ in order to avoid overflow/underflow problems of floating point variables. The increase and attenuation factors are usually set $n^+ \in [1.1, 1.6]$ and $n^- \in [0.4, 0.95]$, respectively. All four parameters are determined using the trial and error approach.

The adaptation mechanism described above has the advantage of correlating the step sizes not to the size of the derivatives but to their signs. Hence, whenever a parameter moves along a direction reducing E (the derivatives at successive epochs have the same sign), its step size is increasing independently of the size of the derivative. In this way, the step sizes can sufficiently increase when needed, even at the final stage of the learning process when the sizes of the derivatives are rather small. Additionally, when changes in the sign of the derivative occur, the step size is diminishing to prevent the error measure from oscillating. As mentioned in [20], due to the temporal relations existing in a dynamic system, the extraction of the ordered partial derivatives is not straightforward and is accomplished via a set of recursive equations. Calculation of the error gradients is based on the use of Lagrange multipliers, as shown below:

$$\frac{\partial^+ E}{\partial b_{2i-1}} = \sum_{k=1}^{k_f} \left\{\lambda_{x,2i-1}(k) \cdot u(k) \cdot f'_a(k, 2i-1)\right\} \quad (8a)$$

$$\frac{\partial^+ E}{\partial b_{2i}} = \sum_{k=1}^{k_f} \left\{\lambda_{x,2i}(k) \cdot u(k) \cdot f'_a(k, 2i)\right\} \quad (8b)$$

$$\frac{\partial^+ E}{\partial c_{2i-1}} = \sum_{k=1}^{k_f} \left\{\lambda_y(k) \cdot x_{2i-1}(k) \cdot f'_b(k)\right\} \quad (9a)$$

$$\frac{\partial^+ E}{\partial c_{2i}} = \sum_{k=1}^{k_f} \left\{\lambda_y(k) \cdot x_{2i}(k) \cdot f'_b(k)\right\} \quad (9b)$$

$$\frac{\partial^+ E}{\partial w_i^{(1)}} = \sum_{k=1}^{k_f} \left\{\lambda_{x,2i-1}(k) \cdot x_{2i-1}(k-1) \cdot f'_a(k, 2i-1) + \right.$$
$$\left. + \lambda_{x,2i}(k) \cdot x_{2i}(k-1) \cdot f'_a(k, 2i)\right\} \quad (10a)$$

$$\frac{\partial^+ E}{\partial w_i^{(2)}} = \sum_{k=1}^{k_f} \left\{\lambda_{x,2i-1}(k) \cdot x_{2i}(k-1) \cdot f'_a(k, 2i-1) - \right.$$
$$\left. - \lambda_{x,2i}(k) \cdot x_{2i-1}(k-1) \cdot f'_a(k, 2i)\right\} \quad (10b)$$

with the Lagrange multipliers derived as follows:

$$\lambda_y(k) = \frac{2}{k_f} \cdot (y(k) - y_d(k)) \quad (11)$$

$$\lambda_{x,2i-1}(k) = \lambda_y(k) \cdot c_{2i-1} \cdot f'_b(k) + $$
$$+ \lambda_{x,2i-1}(k+1) \cdot w_i^{(1)} \cdot f'_a(k+1, 2i-1) - $$
$$- \lambda_{x,2i}(k+1) \cdot w_i^{(2)} \cdot f'_a(k+1, 2i) \quad (12a)$$

$$\lambda_{x,2i}(k) = \lambda_y(k) \cdot c_{2i} f'_b(k) + $$
$$+ \lambda_{x,2i-1}(k+1) \cdot w_i^{(2)} \cdot f'_a(k+1, 2i-1) + $$
$$+ \lambda_{x,2i}(k+1) \cdot w_i^{(1)} \cdot f'_a(k+1, 2i) \quad (12b)$$

where $f'_a(k+1, i)$ and $f'_b(k)$ are the derivatives of $x_i(k+1)$ and $y(k)$, with respect to their arguments. Equations (12) are backward difference equations that can be solved for $k = k_f - 1, \ldots, 1$ using the boundary conditions:

$$\lambda_{x,2i-1}(k_f) = \lambda_y(k_f) \cdot c_{2i-1} \cdot f'_b(k_f) \quad (13a)$$

$$\lambda_{x,2i}(k_f) = \lambda_y(k_f) \cdot c_{2i} \cdot f'_b(k_f) \quad (13b)$$

It should be noted that the learning method is developed for the SISO Scaled Orthogonal BDRNN. Nevertheless, the suggested method is general, also applicable to multi–input–multiple–output BDRNN's of any form, provided that slight modifications are made, based on the structural differences.

3 Experimental results

3.1 Data presentation accuracy measures

The call data come from the Call Detail Records (CDR) of the Private Branch Exchange (PBX) of a large University with more than 6.000 employees and 70.000 students, and an extended telecommunications infrastructure with more than 5.500 telephones. The data set covers a period of 10 years, January 1998 to December 2007, and consists of the monthly calls to national and mobile destinations. Calls to national destinations comprise almost half the volume of the total outgoing calls from the campus. On the other hand, calls to mobile destinations are subject to higher tariffs and they demonstrate an increasing trend during the last fifteen years. According to the International Telecommunication Union (ITU), mobile services

attained a penetration rate of 119.12% for Greece in 2009 [21].

The data set is divided to two subsets: The training set, which is used to perform parameter tuning, and the testing set, which is used for the evaluation of the forecasts. The training set is chosen to be 9 years long (108 observations) and the validation set 1 year long (12 observations) . Due to the variation of days belonging in different months, i.e. February has 28 while January has 31 days, all data are normalized according to (X_k is the actual value):

$$y_d(k) = X_k \cdot \frac{365.25/12}{\text{no of days in month } k} \tag{14}$$

Due to its recurrent nature, the forecaster is trained in parallel mode. Since the testing set is not used in model fitting, the testing set's forecasts are genuine forecasts and can be used to evaluate the forecasting ability of each model. The forecasting accuracy can be evaluated by means of three accuracy measures (k_f is the size of the data set and $y(k)$ is the forecaster's output). The smaller value of each statistic indicates the better fit of the method to the observed data.

–The Root Mean Squared Error (RMSE):

$$E = \sqrt{\frac{1}{k_f} \cdot \sum_{k=1}^{k_f} (y(k) - y_d(k))^2} \tag{15a}$$

–The Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{100}{k_f} \cdot \sum_{k=1}^{k_f} \left| \frac{y_d(k) - y(k)}{y_d(k)} \right| \tag{15b}$$

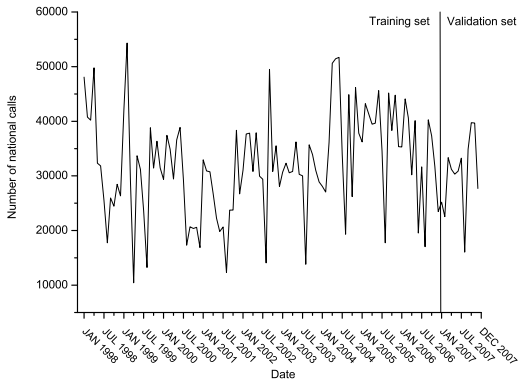
–The Theil's U-statistic, which allows a relative comparison of formal methods with input–output approaches and also squares the errors involved, so that large errors are given much more weight than small errors. It is given by:

$$U = \frac{\sqrt{\sum_{k=1}^{k_f-1} \left(\frac{y(k+1) - y_d(k+1)}{y_d(k)} \right)^2}}{\sqrt{\sum_{k=1}^{k_f-1} \left(\frac{y_d(k+1) - y_d(k)}{y_d(k)} \right)^2}} \tag{15c}$$

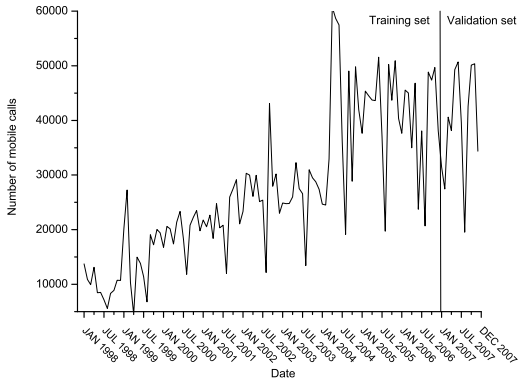
The time series of national and mobile calls are hosted in Fig. 2a and 2b, respectively. It is evident that there exists a distinct seasonal pattern, which is made prevalent from the minimum that occurs in August. Moreover, the number of calls to mobile destinations shows an increasing trend which comports with reports on mobile services penetration [21].

3.2 Comparative analysis

In order to investigate whether the recurrent neural system is capable of discovering the temporal



(a)



(b)

Fig. 2: Monthly number of outgoing calls to (a) national and (b) mobile destinations.

dependencies of both time–series of outgoing calls, a single–input–single–output structure has been selected. The input is the number of the national/mobile calls of the previous month ($u(k) = y_d(k-1)$).

Several BDRNN's with different structural characteristics are examined and the selection of the final model–parameter combination is based on the criteria of (a) effective prediction of the call volume and (b) forecasters of moderate complexity. The selected structural characteristics and the learning parameters' values of the final models are given in Table 1.

In order to compare the proposed forecaster with existing established forecasters that were applied to this particular problem in the past, a comparative analysis with fourteen other models is conducted, on the basis of the accuracy measures mentioned in Subsection 3.1. Three of the models come from the field of computational intelligence (the static fuzzy system described in [8] and the recurrent neurofuzzy systems LR–NFFS and ReNFFOR), while the other eleven are well–established

Table 1: Structural characteristics and learning parameters

Type of calls	No of blocks	$n+$	$n-$	$\Delta_{\min}$	$\Delta_{\max}$	Δ_o	Epochs
National	5	1.5	0.5	1E–4	0.5	0.01	1000
Mobile	5	1.1	0.5	1E–4	0.5	0.05	1000

Table 2: Comparative analysis (testing data set)

Type of calls	National calls			Mobile calls		
Model	RMSE	MAPE	Theils U	RMSE	MAPE	Theils U
Proposed Forecaster	**3293**	**10.461**	**0.261**	**5020**	**12.286**	0.381
LR–NFFS	4367	12.118	0.301	5742	13.166	**0.326**
ReNFFOR	5102	14.890	0.380	7368	19.67	0.452
OLS–FM	4385	13.580	0.366			
NF1	8914	23.846	1.000	12009	28.875	1.000
LESA–M	8570	24.391	0.722	9915	23.046	0.747
LESA–ADD	8418	24.798	0.713	10271	27.218	0.699
SES	6748	20.943	0.515	9671	24.698	0.569
Holts Linear	6753	27.552	0.506	11191	35.507	0.663
Winter's MS	7120	18.415	0.578	9114	20.475	0.665
Winter's AS	6903	17.741	0.553	8495	21.875	0.573
Damped NoS	6862	21.422	0.512	11962	31.756	0.715
Damped MS	7080	19.072	0.573	7419	15.958	0.524
Damped AS	7194	19.838	0.571	9020	23.584	0.599
SARIMA	6064	15.959	0.513	10102	20.793	0.775

statistical forecasting models. The results for each one of these models are presented in Table 2; bold numbers indicate best fit. The performance of the competing rivals is taken from the corresponding references.

The best fit models for each call data category are depicted in the following plots. We choose to present the forecasts produced by the proposed BDRNN model, along with its closest fuzzy/neurofuzzy and its closest classic competitors.

In Fig. 3 the reader may see a comparison for the best fit models for the case of national calls. The plot reveals that the proposed model follows the variation of the actual time series closer than its competitors, a fact that is also evident from its better performance statistics. In the same plot the 95% upper (UCL) and lower (LCL) confidence levels are depicted. These were estimated during the SARIMA fitting process. It should also be stressed that all three forecasts fit well within these confidence intervals and would bear scrutiny with even tighter confidence.

According to Fildes et al., damped trend models are considered "a benchmark forecasting method for all others to beat" [22]. In this sense the proposed BDRNN model exhibits exceptional performance. Visual observation of Fig. 4 reveals the differences between the proposed forecaster and its best rivals for the case of mobile calls. The BDRNN gives better forecast, in the sense that it follows the pattern of the evolution of the series more closely as it identifies all minima and maxima. The 95% confidence intervals for the forecasts

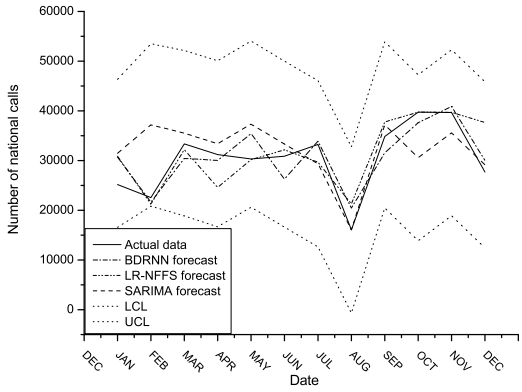


Fig. 3: Comparison of the forecasting ability of the proposed BDRNN forecaster with the best representatives of the two rival model families and the observed number of calls to national destinations. 95% confidence interval is also depicted.

were estimated during the fitting process of the Damped MS model.

4 Conclusions

In this paper a recurrent neural network has been proposed and has been evaluated by having been applied

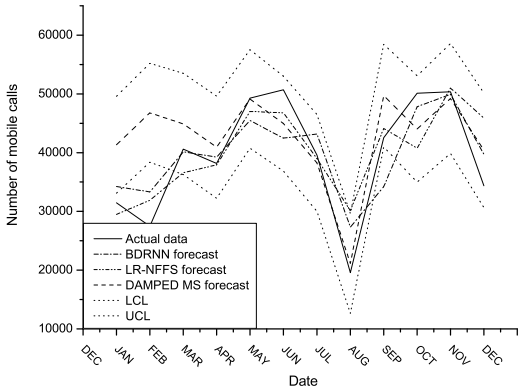


Fig. 4: Comparison of the forecasting ability of the proposed BDRNN forecaster with the best representatives of the two rival model families and the observed number of calls to mobile destinations. 95% confidence interval is also depicted.

on real world telecommunications data. The forecaster is a two–layer network, whose hidden layer consists of pairs of interconnected neurons with internal feedback. Its modelling qualities have been investigated through a comparative analysis with a series of well–established forecasting models and recent fuzzy and neurofuzzy forecasters.

Two different types of calls, according to their destination, have been examined. These are calls to national and calls to mobile destinations. They were chosen because the former represents more than 50% of the outgoing call volume and the later corresponds to more than 1/3 of the telecommunications costs for the organization under study.

According to the results, the proposed forecaster is a promising computational intelligence's approach to the problem of telecommunications call volume forecasting, since it is capable of capturing the time–series dynamics through its internal feedback connections. Therefore it does not require prior knowledge of the order of the time–series or computationally expensive input selection algorithms. Moreover, its structure is quite simpler than those of fully recurrent networks or networks with output feedback. From the analysis it is concluded that it can cope with the problem better than its rivals, either the traditional forecasting methods or the computational intelligence ones.

Forecasting may act as a valuable tool for telecommunications managers and is used for network traffic management, infrastructure optimization and planning, and the scenario planning process. Making use of historical data, managers may predict future demand by creating a reasonably accurate forecast of the call volume. This may be used to make educated strategic decisions on how to charge and bill telecommunications services in their organization, either for profit maximization or for unnecessary cost reduction.

Acknowledgement

The authors would like to thank the members of the Aristotle University's Telecommunications Center for their contribution of anonymised data. The authors wish to acknowledge financial support provided by the Research Committee of the Technological Education Institute of Serres under grant SAT/IC/070213-15/10.

References

[1] S. Makridakis, S. Wheelwright, V. McGee, Forecasting: Methods and Applications, 3rd ed., Wiley, New York, (1998).
[2] C. Hilas, S. Goudos, J. Sahalos, Seasonal Decomposition and Forecasting of Telecommunication Data: A comparative Case Study, Technological Forecasting & Social Change, **73**, 495-509 (2006).
[3] C. Holt, Forecasting Trends and Seasonals by Exponentially Weighted Moving Averages, ONR Memorandum, 52 Carnegie Institute of Technology, Pittsburg (1957). Reprinted: International Journal of Forecasting, **20**, 5-10 (2004).
[4] P. Winters, Forecasting Sales by Exponentially Weighted Moving Averages, Management Science, **6**, 324-342 (1960).
[5] E. Gardner Jr., Exponential Smoothing: The State of the Art, Journal of Forecasting, **4**, 1-28 (1985).
[6] G. Box, G. Jenkins, Time Series Analysis: Forecasting and Control, 2nd ed., Holden-Day, San Francisco, (1976).
[7] G. Madden, T. Joachim, Forecasting Telecommunications Data with Linear Models, Telecommunications Policy, **31**, 31-44 (2007).
[8] P. Mastorocostas, C. Hilas, S. Dova, D. Varsamis, Forecasting of Telecommunications Time-series via an Orthogonal Least Squares-based Fuzzy Model, Proceedings of 21st IEEE International Conference on Fuzzy Systems, (2012).
[9] P. Mastorocostas, C. Hilas, A Computational Intelligence Forecasting System for Telecommunications Time Series, Engineering Applications of Artificial Intelligence **25**, 200-206 (2012).
[10] P. Mastorocostas, C. Hilas, ReNFFor: A Recurrent Neurofuzzy Forecaster for Telecommunications Data, Neural Computing and Applications, (2012).
[11] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modelling and control, IEEE Transactions on Systems, Man, and Cybernetics, **15**, 116-132 (1985).
[12] S. Sivakumar, W. Robertson, W. Phillips, On-Line Stabilization of Block-Diagonal Recurrent Neural Networks, IEEE Transactions on Neural Networks, **10**, 167-175 (1999).
[13] A. Tsoi, A. Back, Locally Recurrent Globally Feedforward Networks: A Critical Review of Architectures, IEEE Transactions on Neural Networks, **5**, 229-239 (1994).
[14] P. Frasconi, M. Gori, and G. Soda, Local feedback multilayered networks, Neural Computation, **4**, 120-130 (1992).

[15] P. Mastorocostas, J. Theocharis, A Stable Learning Method for Block-Diagonal Recurrent Neural Networks: Application to the Analysis of Lung Sounds, IEEE Transactions on Systems, Man, and Cybernetics Part B, **36**, 242-254 (2006).
[16] M. Riedmiller, H. Braun, A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm, Proceedings of IEEE International Joint Conference on Neural Networks, 586-591 (1993).
[17] C. Igel, M. Husken, Empirical Evaluation of the Improved RPROP Learning Algorithms, Neurocomputing, **50**, 105-123 (2003).
[18] P. Mastorocostas, Resilient Back Propagation Learning Algorithm for Recurrent Fuzzy Neural Networks, IET Electronics Letters, **40**, 57-58 (2004).
[19] P. Werbos, Beyond Regression: new tools for prediction and analysis in the behavioral sciences, Ph.D. Thesis, Harvard Univ., Cambridge, MA (1974).
[20] S. Piche, Steepest Descent Algorithms for Neural Network Controllers and Filters, IEEE Transactions on Neural Networks, **5**, 198-212 (1994).
[21] ITU-T ICTeye Reports, Available online : http://www.itu.int/ITU-D/ICTEYE/Reports.aspx, (2010).
[22] R. Fildes, K. Nikolopoulos, S. Crone, A. Syntetos, Forecasting and Operational Research: a Review, Journal of the Operational Research Society, **59**, 1150-1172 (2008).

Paris Mastorocostas was born in Thessaloniki, Greece, in 1971. He received the Diploma and Ph.D. degrees in Electrical & Computer Engineering from Aristotle University of Thessaloniki, Greece. Presently he serves as Professor at the Dept. of Informatics & Communications, Technological Education Institute of Serres, Greece. His research interests include fuzzy and neural systems with applications to identification and classification processes, data mining and scientific programming.

Constantinos Hilas was born in Thessaloniki, Greece, in 1967. He received the B.Sc., M.Sc. and Ph.D. degrees from the Dept. of Physics, Aristotle University of Thessaloniki, Greece, and the M.Sc. degree in Information Systems from the University of Macedonia, Greece. Presently he serves as Assistant Professor at the Dept. of Informatics & Communications, Technological Education Institute of Serres, Greece. His research interests include the application of data mining techniques on communications & networking problems, forecasting, user behavior modeling and profiling, classification and characterization.

Dimitris Varsamis was born in Serres, Greece, in 1976. He received the B.Sc. degree in Mathematics from the Dept. of Mathematics, University of Ioannina, Greece, the M.Sc. in Theoretical Computer Science, Control and Systems Theory, and the Ph.D. degrees from the Dept. of Mathematics, Aristotle University of Thessaloniki, Greece. Presently he serves as Lecturer at the Dept. of Informatics & Communications, Technological Education Institute of Serres, Greece. His research interests include scientific programming, computational methods, control system theory and numerical analysis.

Stergiani Dova was born in Thessaloniki, Greece, in 1988. She received the B.Sc. degree from the Dept. of Informatics & Communications, Technological Education Institute of Serres, Greece. Presently, she is completing her M.Sc. in Information Systems at the University of Macedonia, Greece. Her research interests include fuzzy systems, neural networks, algorithms analysis and scientific computing.